This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

PORTAL
US Patent & Trademark Office

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ◉ The ACM Digital Library ○ The Guide

critical process monitor and periodic and event

Search

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used **critical process monitor** and **periodic** and **event** Found **77,387** of **143,484**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

☐ Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown Relevance scale

1 An approach to reducing delays in recognizing distributed event occurrences

Madalene Spezialetti

December 1991 **ACM SIGPLAN Notices , Proceedings of the 1991 ACM/ONR workshop on Parallel and distributed debugging**, Volume 26 Issue 12

Full text available: pdf(1.23 MB) Additional Information: full citation, references, citings, index terms

2 A real-time monitor for a distributed real-time operating system

Hideyuki Tokuda, Makoto Kotera, Clifford E. Mercer

November 1988 **ACM SIGPLAN Notices , Proceedings of the 1988 ACM SIGPLAN and SIGOPS workshop on Parallel and distributed debugging**, Volume 24 Issue 1

Full text available: pdf(967.71 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Monitoring and debugging for a distributed real-time system is a complicated problem due to the lack of a set of advanced tools and adequate operating system capability. Software tools can cover the wide range of the software development life cycle from the requirement analysis phase to debugging and maintenance phases. However, many of these modern tools are not effective for building or analyzing complex real-time systems. Real-time software tools and effective kernel support are essentia ...

3 Paradigms for process interaction in distributed programs

Gregory R. Andrews

March 1991 **ACM Computing Surveys (CSUR)**, Volume 23 Issue 1

Full text available: pdf(3.77 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Distributed computations are concurrent programs in which processes communicate by message passing. Such programs typically execute on network architectures such as networks of workstations or distributed memory parallel machines (i.e., multicomputers such as hypercubes). Several paradigms—examples or models—for process interaction in distributed computations are described. These include networks of filters, clients, and servers, heartbeat algorithms, probe/echo algorithms, broa ...

Keywords: clients and servers, distributed and parallel algorithms, distributed programming, distributed programming methods, heartbeat algorithms, networks of filters, patterns for interprocess communication, probe/echo algorithms, replicated servers, token-passing algorithms

4 Extracting usability information from user interface events
David M. Hilbert, David F. Redmiles
December 2000 **ACM Computing Surveys (CSUR)**, Volume 32 Issue 4

Full text available: pdf(1.50 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Modern window-based user interface systems generate user interface events as natural products of their normal operation. Because such events can be automatically captured and because they indicate user behavior with respect to an application's user interface, they have long been regarded as a potentially fruitful source of information regarding application usage and usability. However, because user interface events are typically voluminos and rich in detail, automated support is generally ...

Keywords: human-computer interaction, sequential data analysis, usability testing, user interface event monitoring

5 Monit: a performance monitoring tool for parallel and pseudo-parallel programs
Teemu Kerola, Herb Schwetman
May 1987 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1987 ACM SIGMETRICS conference on Measurement and modeling of computer systems**, Volume 15 Issue 1

Full text available: pdf(969.76 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes a performance monitoring system, Monit, developed for performance evaluation of parallel systems. Monit uses trace files that are generated during the execution of parallel programs. Monit analyzes these trace files and produces time-oriented graphs of resource usage and system queues. Users interactively select the displayed items, resolution, and time intervals of interest. The current implementation of Monit is for SUN-3 workstation, but the program is easily adaptab ...

6 Session S8.1: power and battery management: Process cruise control: event-driven clock scaling for dynamic power management
Andreas Weissel, Frank Bellosa
October 2002 **Proceedings of the 2002 international conference on Compilers, architecture, and synthesis for embedded systems**

Full text available: pdf(161.01 KB) Additional Information: full citation, abstract, references, index terms

Scalability of the core frequency is a common feature of low-power processor architectures. Many heuristics for frequency scaling were proposed in the past to find the best trade-off between energy efficiency and computational performance. With complex applications exhibiting unpredictable behavior these heuristics cannot reliably adjust the operation point of the hardware because they do not know where the energy is spent and why the performance is lost.Embedded hardware monitors in the form of ...

Keywords: clock scaling, event counters, power management, scheduling

7 A simple student environment for lightweight process concurrent programming under SunOs
Toby S. Berk
March 1996 **ACM SIGCSE Bulletin , Proceedings of the twenty-seventh SIGCSE technical symposium on Computer science education**, Volume 28 Issue 1

Full text available: pdf(471.40 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes a simplified student environment for lightweight processes (threads) under SunOs or compatible systems. The environment is built on top of the SunOs lwp package. A range of concurrency facilities are provided including critical sections, events and signals, and monitors. A hidden scheduler enhances the effects of the nondeterminacy, even for small student programs. A sample assignment is shown using the package.

8 The flight recorder: an architectural aid for system monitoring
Michael M. Gorlick
December 1991 **ACM SIGPLAN Notices , Proceedings of the 1991 ACM/ONR workshop on Parallel and distributed debugging**, Volume 26 Issue 12
Full text available: pdf(944.95 KB) Additional Information: full citation, references, citings, index terms

9 Guide for the use of the Ada Ravenscar Profile in high integrity systems
Alan Burns, Brian Dobbing, Tullio Vardanega
June 2004 **ACM SIGAda Ada Letters**, Volume XXIV Issue 2
Full text available: pdf(548.17 KB) Additional Information: full citation, references

10 Software safety: why, what, and how
Nancy G. Leveson
June 1986 **ACM Computing Surveys (CSUR)**, Volume 18 Issue 2
Full text available: pdf(4.18 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Software safety issues become important when computers are used to control real-time, safety-critical processes. This survey attempts to explain why there is a problem, what the problem is, and what is known about how to solve it. Since this is a relatively new software research area, emphasis is placed on delineating the outstanding issues and research topics.

11 Experience with processes and monitors in Mesa
Butler W. Lampson, David D. Redell
February 1980 **Communications of the ACM**, Volume 23 Issue 2
Full text available: pdf(1.22 MB) Additional Information: full citation, references, citings

Keywords: concurrency, condition, deadlock, module, monitor, operating system, process, synchronization, task, variable

12 Specification and implementation of exceptions in workflow management systems
Fabio Casati, Stefano Ceri, Stefano Paraboschi, Guiseppe Pozzi
September 1999 **ACM Transactions on Database Systems (TODS)**, Volume 24 Issue 3
Full text available: pdf(250.40 KB) Additional Information: full citation, abstract, references, citings, index terms

Although workflow management systems are most applicable when an organization follows standard business processes and routines, any of these processes faces the need for handling exceptions, i.e., asynchronous and anomalous situations that fall outside the normal control flow.In this paper we concentrate upon anomalous situtations that, although

unusual, are part of the semantics of workflow applications, and should be specified and monitored coherently; in most real-life applica ...

Keywords: active rules, asynchronous events, exceptions, workflow management systems

13 An alternative to event queues for synchronization in monitors

J. L. W. Kessels

July 1977 **Communications of the ACM**, Volume 20 Issue 7

Full text available: pdf(372.51 KB) Additional Information: full citation, abstract, references, citings

In the monitor concept, as proposed by Brinch Hansen and Hoare, event queues are used for synchronization. This paper describes another synchronizing primitive which is nearly as expressive as the conditional wait, but can be implemented more efficiently. An implementation of this primitive in terms of P and V operations is given together with a correctness proof. Two examples are presented: the readers and writers problem and the problem of information streams sharing a finite buffer pool. ...

Keywords: conditional critical region, monitor, mutual exclusion, operating system, structuring concept, synchronization

14 Event-based debugging of object/action programs

Chu-Chung Lin, Richard J. LeBlanc

November 1988 **ACM SIGPLAN Notices , Proceedings of the 1988 ACM SIGPLAN and SIGOPS workshop on Parallel and distributed debugging**, Volume 24 Issue 1

Full text available: pdf(1.25 MB) Additional Information: full citation, references, citings, index terms

15 OMEGA: a language and system for on-line monitoring of software-intensive processes

Sorana Cîmpan, Flavio Oquendo

July 2000 **ACM SIGSOFT Software Engineering Notes**, Volume 25 Issue 4

Full text available: pdf(1.22 MB) Additional Information: full citation, abstract, index terms

This paper presents an environment for monitoring software-intensive processes: the *Omega* environment (Omega stands for *On-line Monitoring Environment: General and Adaptable*). The environment provides the language *Omega/MDL* (*Monitoring Definition Language*) for defining monitoring models as well as a mechanism for the execution of such models *Omega/EM* (*Execution Mechanism*). The executing monitoring models (i.e. monitori ...

Keywords: environment, fuzzy logic, monitoring, software engineering, software-intensive processes

16 The impact of object technology on commercial transaction processing

Edward E. Cobb

August 1997 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 6 Issue 3

Full text available: pdf(649.17 KB) Additional Information: full citation, abstract, index terms

Businesses today are searching for information solutions that enable them to compete in the global marketplace. To minimize risk, these solutions must build on existing investments, permit the best technology to be applied to the problem, and be manageable. Object technology, with its promise of improved productivity and quality in application

development, delivers these characteristics but, to date, its deployment in commercial business applications has been limited. One possible reason is the ...

Keywords: Objects, Workflow, transaction processing

17 An object-based infrastructure for program monitoring and steering
Greg Eisenhauer, Karsten Schwan
August 1998 **Proceedings of the SIGMETRICS symposium on Parallel and distributed tools**
Full text available: pdf(1.50 MB) Additional Information: full citation, references, index terms

18 APPL/A: a language for software process programming
Stanley M. Sutton, Dennis Heimbigner, Leon J. Osterweil
July 1995 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 4 Issue 3
Full text available: pdf(4.89 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Software process programming is the coding of software processes in executable programming languages. Process programming offers many potential benefits, but their realization has been hampered by a lack of experience in the design and use of process programming languages. APPL/A is a prototype software process programming language developed to help gain this experience. It is intended for the coding of programs to represent and support software processes including process, product, and p ...

Keywords: consistency management, multiparadigm programming languages, software process programming, transaction management

19 Adaptive rate-controlled scheduling for multimedia applications
David K. Y. Yau, Simon S. Lam
February 1997 **Proceedings of the fourth ACM international conference on Multimedia**
Full text available: pdf(1.38 MB) Additional Information: full citation, references, citings, index terms

Keywords: CPU scheduling, QoS guarantee, adaptive rate control, continuous media, firewall property, rate reservation

20 Wide-area monitoring of mobile objects: Implementing software on resource-constrained mobile sensors: experiences with Impala and ZebraNet
Ting Liu, Christopher M. Sadler, Pei Zhang, Margaret Martonosi
June 2004 **Proceedings of the 2nd international conference on Mobile systems, applications, and services**
Full text available: pdf(3.14 MB) Additional Information: full citation, abstract, references, index terms

ZebraNet is a mobile, wireless sensor network in which nodes move throughout an environment working to gather and process information about their surroundings[10]. As in many sensor or wireless systems, nodes have critical resource constraints such as processing speed, memory size, and energy supply; they also face special hardware issues such as sensing device sample time, data storage/access restrictions, and wireless transceiver capabilities. This paper discusses and evaluates ZebraNet's syst ...

Keywords: event handling, middleware system, network communications, operation scheduling, sensor networks

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player